

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-016409

(43)Date of publication of application : 17.01.1997

(51)Int.Cl.

G06F 9/46

(21)Application number : 07-166443

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 30.06.1995

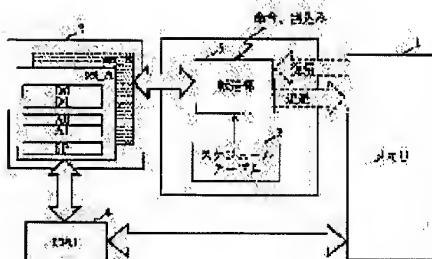
(72)Inventor : MATSUZAKI TOSHIMICHI
HAMAGUCHI TOSHIKUMI
TANASE HIROSHI
MATSUMOTO MASAHICO

(54) MICROCOMPUTER

(57)Abstract:

PURPOSE: To provide the microcomputer which switches tasks fast without increasing the cost.

CONSTITUTION: A CPU 4 switches the tasks according to the execution order in a task schedule table 2 and executes the tasks by using one register set of a register file 3. Each time a task is switched, the register sets used for the task execution by the CPU 4 are alternated. Each time the task is switched, a transmission part 5 saves and reloads the register sets by utilizing free bus cycles of the CPU 4. Consequently, the fast task switching can be performed only by adding one register set and one transfer means as hardware.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-16409

(43)公開日 平成9年(1997)1月17日

(51)Int.Cl.⁶
G 0 6 F 9/46

識別記号
3 1 3

府内整理番号
F I
G 0 6 F 9/46

3 1 3 D

技術表示箇所

審査請求 未請求 請求項の数6 O.L (全14頁)

(21)出願番号

特願平7-166443

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(22)出願日

平成7年(1995)6月30日

(72)発明者 松崎 敏道

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 浜口 敏文

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 多那瀬 寛

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74)代理人 弁理士 中島 司朗

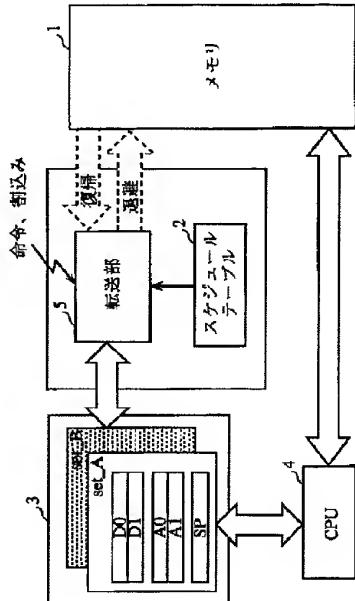
最終頁に続く

(54)【発明の名称】 マイクロコンピュータ

(57)【要約】

【目的】コストを増大させることなく高速にタスクを切り換えるマイクロコンピュータを提供することを目的とする。

【構成】CPU4は、タスクスケジュールテーブル2の実行順序に従ってタスクを切り換える、レジスタファイル3の一方のレジスタセットを使用してタスクを実行する。また、タスク切り換え毎に、CPU4のタスク実行に使用されるレジスタセットが交互に切り換えられる。タスク切り換え毎に、転送部5は、CPU4の空バスサイクルを利用して、それぞれレジスタセット退避、復帰を行う。これにより、レジスタセットを1つ分と、転送手段をハードウェアとして追加するだけで、高速なタスク切り換えを実現することができる。



1

【特許請求の範囲】

【請求項1】複数のタスクを順に切り換えるながら実行するマイクロコンピュータであって、同一のレジスタ構成をもち、タスク実行に一方が使用される2つのレジスタセットと、タスク切り換えと同時に、タスク実行に使用されるレジスタセットを他方に切り換える切り換え手段と、レジスタセットに接続された内部バス、およびメモリに接続されたメモリバスそれぞれの空サイクルを検出する検出手段と、タスク切り換え毎に、検出されたバスの空サイクルを利用して、タスク実行に使用されていないレジスタセットのデータをメモリに退避し、次の切り換え後に実行すべきタスク用のデータをメモリから当該レジスタセットに復帰する転送手段とを備えることを特徴とするマイクロコンピュータ。

【請求項2】前記転送手段は、内部バスの空サイクルにおいて、タスク実行に使用されていないレジスタセットのレジスタデータを読み出すレジスタ読み出し手段と、メモリバスの空サイクルにおいて、レジスタ読み出し手段により読み出されたデータをメモリに書き込むメモリ書き込み手段と、メモリ書き込み手段による書き込み完了後、メモリバスの空サイクルにおいて、次のタスク用のデータをメモリから読み出すメモリ読み出し手段と、内部バスの空サイクルにおいて、メモリ読み出し手段により読み出されたデータを、タスク実行に使用されていないレジスタセットに書き込むレジスタ書き込み手段とを有することを特徴とする請求項1記載のマイクロコンピュータ。

【請求項3】複数のタスクを順に切り換えるながら実行するマイクロコンピュータであって、複数のタスクプログラムを記憶し、各タスクに対応するデータ退避領域を有するメモリと、複数のタスクの実行順序を記憶するテーブル手段と、同一のレジスタ構成をもつ2つのレジスタセットと、テーブル手段の実行順序に従ってタスクを切り換え、レジスタセットの一方を使用してタスクを実行する中央演算処理手段と、

タスク切り換え毎に、中央演算処理手段のタスク実行に使用されるレジスタセットを交互に切り換える切り換え手段と、レジスタセットに接続された内部バス、およびメモリに接続されたメモリバスそれぞれの空サイクルを検出する検出手段と、タスク切り換え毎に、検出されたバスの空サイクルを利用して、タスク実行に使用されていないレジスタセットのデータを、切り換え前のタスクに対応するデータ退避領域に退避する退避手段と、

2

退避手段による退避後、検出されたバスの空サイクルを利用して、中央演算処理手段による実行中タスクの次のタスクに対応するデータ退避領域のデータを、タスク実行に使用されていないレジスタセットに復帰させる復帰手段とを備えることを特徴とするマイクロコンピュータ。

【請求項4】前記メモリは、各タスク毎のスタック領域内に前記データ退避領域を有し、前記テーブル手段は、タスクの実行順序と、各タスクに

10 対応するスタックポインタの内容とを記憶することを特徴とする請求項3記載のマイクロコンピュータ。

【請求項5】前記退避手段は、内部バスの空サイクルにおいて、タスク実行に使用されていないレジスタセットのレジスタデータを順に読み出すレジスタ読み出し手段と、

メモリバスの空サイクルにおいて、レジスタ読み出し手段により読み出されたデータを順に、切り換え前のタスクに対応するデータ退避領域に書き込むメモリ書き込み手段とを有し、

20 前記復帰手段は、メモリ書き込み手段による書き込み完了後、メモリバスの空サイクルにおいて、中央演算処理手段による実行中タスクの次のタスクに対応するデータ退避領域のデータを順にメモリから読み出すメモリ読み出し手段と、内部バスの空サイクルにおいて、メモリ読み出し手段により読み出されたデータを、タスク実行に使用されていないレジスタセットに書き込むレジスタ書き込み手段とを有することを特徴とする請求項3又は4記載のマイクロコンピュータ。

30 【請求項6】前記マイクロコンピュータは、さらに、中央演算処理手段に動作クロックを供給するクロック供給手段と、復帰手段による復帰動作中に、中央演算処理手段によりタスク切り換えが発生した場合に、当該復帰動作が完了するまでの間クロック供給手段の動作クロックの供給を停止する停止手段とを備えることを特徴とする請求項3、4又は5記載のマイクロコンピュータ。

【発明の詳細な説明】

【0001】

40 【産業上の利用分野】本発明は、複数のタスクを切り換えるながら実行するマイクロコンピュータに関する。

【0002】

【従来の技術】近年、マイクロコンピュータは、あらゆる機器に利用されており、処理能力の向上に伴ってより広い分野でより複雑な制御に応用されている。組み込み用途のマイクロコンピュータにおいても、より複雑な制御に対応できるように、マルチタスク管理などより高度な機能が望まれている。

【0003】図6は、第1の従来技術におけるマルチタスク機能を有するコンピュータの説明図である。同図に

50

においてCPU10は、演算部11とレジスタセット12とからなる。演算部11は、タスクを1つずつ実行する。レジスタセット12は、タスク実行に要するデータを保持する。

【0004】メモリ13は、各タスク毎にコンテキスト(レジスタセットのデータ、PCの内容、PSWの内容等)を退避する領域であるコンテキストブロック14を有する。同図では、コンテキストブロック14中にタスク1～3用のコンテキスト退避領域が示されている。この第1の従来例におけるタスクの管理はソフトウェア(OS)によって行われる。例えばタスク1からタスク2に切り換える場合、OSは、タスク切り換えを指示するイベント(タイム割込み等)を受け付けると、まずCPU10からメモリ13にタスク1のコンテキストを退避し、次にメモリ13からCPU10にタスク2のコンテキストを復帰し、さらにタスク2の実行を開始する。

【0005】上記のように第1の従来例では、ソフトウェア(OS)によりマルチタスクを実現しているので、ハードウェア規模が小さく済むという利点がある。図7は、第2の従来技術におけるマルチタスク機能を有するコンピュータの説明図である。同図においてCPU20は、演算部21、レジスタファイル22、レジスタセット切り換え部23、制御部24とからなる。

【0006】演算部21は、タスクを1つずつ実行する。レジスタファイル22は、タスク実行に要するデータを保持する複数のレジスタセットからなる。レジスタセット切り換え部23は、現在実行中のタスクに対応するレジスタセットを有効にする。この有効なレジスタセットのみが演算部21に使用される。

【0007】制御部24は、タスクの切り換えを制御する。メモリ25は、レジスタセットのデータを除くコンテキストの退避領域を有する。この第2の従来例におけるタスクの管理はハードウェアによって行われる。例えばタスク1からタスク2に切り換える場合、制御部24は、タスク切り換えを指示するイベント(タイム割込み等)を受け付けると、レジスタセット切り換え部23にタスク2に対応するレジスタセットを有効にするように指示する共に、演算部21のPCの内容、PSWの内容等をメモリ25に退避し、タスク2のPCの内容、PSWの内容等を演算部21に復帰する。その結果、演算部21は、有効なレジスタセット2を用いてタスク2を実行する。

【0008】上記のように第2の従来例では、ハードウェアによりマルチタスクを実現しているので、タスクの切り換えが高速に行えるという利点がある。

【0009】

【発明が解決しようとする課題】しかしながら、上記従来技術によれば、組み込み用途のマイクロコンピュータにおけるマルチタスク管理には適していないという問題があった。詳しくいうと、第1の従来例によれば、タス

ク切り換えに際してコンテキストの退避および復帰をソフトウェア(OS)により実現するので切り換え速度が遅く、リアルタイム性を要求される制御を行う組み込み用途のマイクロコンピュータには適していない。また、第2の従来例によれば、タスク数を同数のレジスタセットを必要とし、タスク切り換え行う制御部をハードウェアにて実現するので、ハードウェア規模が大きくなる。その結果コストが高くなり組み込み用途のマイクロコンピュータには適していない。

10 【0010】本発明は上記の問題点に鑑み、コストを増大させることなく高速にタスクを切り換えるマイクロコンピュータを提供することを目的とする。

【0011】

【課題を解決するための手段】上記課題を解決するため、請求項1の発明は、複数のタスクを順に切り換えるながら実行するマイクロコンピュータであって、同一のレジスタ構成をもち、タスク実行に一方が使用される2つのレジスタセットと、タスク切り換えと同時に、タスク実行に使用されるレジスタセットを他方に切り換える切り換え手段と、レジスタセットに接続された内部バス、およびメモリに接続されたメモリバスそれぞれの空サイクルを検出する検出手段と、タスク切り換え毎に、検出されたバスの空サイクルを利用して、タスク実行に使用されていないレジスタセットのデータをメモリに退避し、次の切り換え後に実行すべきタスク用のデータをメモリから当該レジスタセットに復帰する転送手段とを備える。

20 【0012】請求項2の発明は、請求項1において前記転送手段が、内部バスの空サイクルにおいて、タスク実行に使用されていないレジスタセットのレジスタデータを読み出すレジスタ読み出し手段と、メモリバスの空サイクルにおいて、レジスタ読み出し手段により読み出されたデータをメモリに書き込むメモリ書き込み手段と、メモリ書き込み手段による書き込み完了後、メモリバスの空サイクルにおいて、次に実行すべきタスク用のデータをメモリから読み出すメモリ読み出し手段と、内部バスの空サイクルにおいて、メモリ読み出し手段により読み出されたデータを、タスク実行に使用されていないレジスタセットに書き込むレジスタ書き込み手段とを有している。

30 【0013】請求項3の発明は、複数のタスクを順に切り換えるながら実行するマイクロコンピュータであって、複数のタスクプログラムを記憶し、各タスクに対応するデータ退避領域を有するメモリと、複数のタスクの実行順序を記憶するテーブル手段と、同一のレジスタ構成をもつ2つのレジスタセットと、テーブル手段の実行順序に従ってタスクを切り換える、レジスタセットの一方を使用してタスクを実行する中央演算処理手段と、タスク切り換え毎に、中央演算処理手段のタスク実行に使用されるレジスタセットを交互に切り換える切り換え手段と、

5

レジスタセットに接続された内部バス、およびメモリに接続されたメモリバスそれぞれの空サイクルを検出する検出手段と、タスク切り換え毎に、検出されたバスの空サイクルを利用して、タスク実行に使用されていないレジスタセットのデータを、切り換え前のタスクに対応するデータ退避領域に退避する退避手段と、退避手段による退避後、検出されたバスの空サイクルを利用して、中央演算処理手段による実行中タスクの次のタスクに対応するデータ退避領域のデータを、タスク実行に使用されていないレジスタセットに復帰させる復帰手段とを備える。

【0014】請求項4の発明は、請求項3において、前記メモリは、各タスク毎のスタック領域内に前記データ退避領域を有し、前記テーブル手段は、タスクの実行順序と、各タスクに対応するスタックボインタの内容とを記憶する。請求項5の発明は、請求項3又は4において、前記退避手段は、内部バスの空サイクルにおいて、タスク実行に使用されていないレジスタセットのレジスタデータを順に読み出すレジスタ読み出し手段と、メモリバスの空サイクルにおいて、レジスタ読み出し手段により読み出されたデータを順に、切り換え前のタスクに對応するデータ退避領域に書き込むメモリ書き込み手段とを有し、前記復帰手段は、メモリ書き込み手段による書き込み完了後、メモリバスの空サイクルにおいて、中央演算処理手段による実行中タスクの次のタスクに對応するデータ退避領域のデータを順にメモリから読み出すメモリ読み出し手段と、内部バスの空サイクルにおいて、メモリ読み出し手段により読み出されたデータを、タスク実行に使用されていないレジスタセットに書き込むレジスタ書き込み手段とを有する。

【0015】請求項6の発明は、請求項3、4又は5において、さらに、中央演算処理手段に動作クロックを供給するクロック供給手段と、復帰手段による復帰動作中に、中央演算処理手段によりタスク切り換えが発生した場合に、当該復帰動作が完了するまでの間クロック供給手段の動作クロックの供給を停止する停止手段とを備える。

【0016】

【作用】上記の手段により、請求項1の発明に係るマイクロコンピュータでは、切り換え手段は、タスク切り換えと同時に、タスク実行に使用されるレジスタセットを切り換える。タスク切り換え毎に、転送手段は、検出されたバスの空サイクルを利用して、タスク実行に使用されていないレジスタセットのデータをメモリに退避し、次の切り換え後に実行すべきタスク用のデータをメモリから当該レジスタセットに復帰する。これにより、レジスタセットを1つ分と、転送手段をハードウェアとして追加するだけで、高速なタスク切り換えを実現することができる。

【0017】請求項2の発明に係るマイクロコンピュー

10

タでは、請求項1の前記転送手段において、レジスタ読み出し手段とメモリ書き込み手段とにより退避が行われ、メモリ読み出し手段とレジスタ書き込み手段とにより復帰が行われる。その際、レジスタ読み出し手段と、レジスタ書き込み手段とは内部バスの空サイクルにおいてレジスタにアクセスし、メモリ読み出し手段とメモリ書き込み手段とはメモリバスの空サイクルにおいてメモリにアクセスする。これにより、タスク実行を全く妨げることなく、退避と復帰を実現できる。

【0018】請求項3の発明に係るマイクロコンピュータでは、中央演算処理手段は、テーブル手段の実行順序に従ってタスクを切り換え、レジスタセットの一方を使用してタスクを実行する。切り換え手段は、タスク切り換え毎に、中央演算処理手段のタスク実行に使用されるレジスタセットを交互に切り換える。タスク切り換え毎に、退避手段、復帰手段は、検出されたバスの空サイクルを利用して、それぞれ退避、復帰を行う。これにより、レジスタセットを1つ分と、転送手段をハードウェアとして追加するだけで、高速なタスク切り換えを実現することができる。

【0019】請求項4の発明に係るマイクロコンピュータでは、請求項3において、前記データ退避領域が、メモリ中の各タスク毎のスタック領域内に設けられ、前記テーブル手段は、タスクの実行順序と、各タスクに對応するスタックボインタの内容とを記憶する。これにより、タスクの実行順序として1つのタスクを複数指定するだけで、中央演算処理手段におけるタスクの占有率を簡単に制御できる。

【0020】請求項5の発明に係るマイクロコンピュータでは、請求項3又は4において、レジスタ読み出し手段とメモリ書き込み手段とにより退避が行われ、メモリ読み出し手段とレジスタ書き込み手段とにより復帰が行われる。その際、レジスタ読み出し手段と、レジスタ書き込み手段とは内部バスの空サイクルにおいてレジスタにアクセスし、メモリ読み出し手段とメモリ書き込み手段とはメモリバスの空サイクルにおいてメモリにアクセスする。これにより、タスク実行を全く妨げることなく、退避と復帰を実現できる。

【0021】請求項6の発明に係るマイクロコンピュータでは、停止手段は、復帰手段による復帰動作中に、中央演算処理手段によりタスク切り換えが発生した場合に、当該復帰動作が完了するまでの間クロック供給手段の動作クロックの供給を停止する。これにより、退避・復帰に要するよりも短い周期でタスク切り換えの要求が発生しても適切にタスクを切り換えることができる。タスク切り換え周期を幅広く設定することができる。

【0022】

【実施例】図1は、本発明の実施例におけるマイクロコンピュータの主要な概略構成を示すブロック図である。

50

このマイクロコンピュータは、メモリ1、タスクスケジュールテーブル2、レジスタファイル3、CPU4、転送部5を有する。メモリ1は、複数のタスクからなるプログラムやデータを記憶し、また、各タスク用の退避領域を有する。この退避領域には、レジスタセットの内容、プログラムカウンタの内容、プロセッサステータスワードの内容等を含むコンテキストが退避される。

【0023】タスクスケジュールテーブル2は、物理的にはメモリ1内に設けられ、タスクの実行順序を保持する。レジスタファイル3は、レジスタセットAとレジスタセットBとからなる。この2つのレジスタセットA、Bは、タスクの切り換え毎に交互に切り換えられ、常に一方がCPU4のタスク実行に使用される。

【0024】CPU4は、複数のタスクを順に切り換えながら実行する。CPU4が実行するタスクは、タスク切り換えを指示するイベントが発生する毎に、タスクスケジュールテーブル2の内容に従って切り換えられる。ここでイベントは、例えば、周期的なタイマー割込み、タスクスイッチ命令、特定ポート（シリアルポートなど）割り込み等である。

【0025】転送部5は、タスクが切り換えられる毎に、CPU4によるタスク実行と独立して、CPU4により使用されていないレジスタセット内のデータをメモリに退避し、次に実行すべきタスク用のデータをメモリから当該レジスタセットに復帰する。図2は、図1に示したマイクロコンピュータのより詳細な構成を示すブロック図である。

【0026】同図に示すように、本マイクロコンピュータは、ROM101、RAM102、インクリメンタ401、セレクタ402、命令アドレスバッファ403、命令フェッチバッファ405、プロセッサステータスワード（以下PSWと略す）406、命令バッファ407、命令レジスタ408、ステータスレジスタ409、PLA410、マイクロ命令レジスタ411、ALU412、ALOB413、オペランドアドレスバッファ414、ストアバッファ415、ロードバッファ416、セレクタ417、転送制御部501、PLA502、マイクロ命令レジスタ503、TSTB504、TSTP505、オペランドアドレスバッファ506、ストアバッファ507、ロードバッファ508、加算器509、定数発生器510とから構成され、ROM101およびRAM102は、図1に示したメモリ1に相当し、複数のタスクを含むプログラム、タスクスケジュールテーブル2、タスク用退避領域を有する。このプログラムはROM101に記憶され、タスクスケジュールテーブル2は、ROM101またはRAM102に設けられ、タスク用退避領域はRAM102に設けられる。

【0027】レジスタセット301とレジスタセット302とは、同一のレジスタ構成であり、それぞれデータレジスタ（D1、D0）と、アドレスレジスタ（A1、

A0）と、スタックポインタ（SP）と、これらからのデータを選択的に出力するセレクタとを有する。インクリメンタ401、セレクタ402、命令アドレスバッファ403、命令フェッチバッファ405からなる回路部分は、ROM101から命令を順時ブリフェッチする命令準備部である。

【0028】ステータスレジスタ406は、CPU4の動作状態を表す各種フラグを有するレジスタである。命令バッファ407からなる回路部分は、命令準備部によりブリフェッチされた複数の命令を保持する命令キューである。命令レジスタ408、ステータスレジスタ409、PLA410、マイクロ命令レジスタ411からなる回路部分は、命令準備部又は命令キューから供給される命令を解読し、その実行を制御する命令制御部である。また、この命令制御部は、タスク切り換えを指示するイベントが発生する毎に、タスクスケジュールテーブル2の内容に従ってタスクを切り換える制御およびレジスタセットを交互に切り換える制御を行う。本実施例では上記イベントは、タイマー（図外）による200mS

20 每の周期的な割込みとする。また、命令制御部は、内部バス418に対してデータを入出力する制御を行う。

【0029】ALU412、ALOB413、オペランドアドレスバッファ414、ストアバッファ415、ロードバッファ416、セレクタ417からなる回路部分は、命令制御部の制御に従って命令を実行する命令実行部である。内部バス418は、2本のバスからなり、主としてレジスタファイル3に対して読み書きされるデータを伝達する。

【0030】メモリバス419は、アドレスバスとデータバスとからなり、主としてROM101、RAM102、周辺回路に対してアドレス、データを伝達する。バスコントローラ420は、メモリバス419を使用するアクセスが競合する場合、例えば、命令準備部によるブリフェッチと、命令実行部によるオペランドデータのロード／ストアとの競合する場合に、バス使用を調停する。バスコントローラ420は、この調停によりメモリバス419に対してデータを入出力するための各種制御信号を出力する。

【0031】クロック供給部421は、CPU4及び転送部5に対して動作クロックを供する。転送制御部501は、PLA502、マイクロ命令レジスタ503からなり、命令制御部によりタスクの切り換えが行われた直後に、CPU4により使用されていない方のレジスタセットの内容を、切り換えの直前に実行されていたタスクの退避領域に退避し、さらに、現在実行されているタスクの次に実行すべきタスクの退避領域の内容を当該レジスタセットに復帰する制御を行う。

【0032】PLA502は、前記退避及び復帰を実現するマイクロ命令を発行する。マイクロ命令レジスタ503は、PLA502からのマイクロ命令に対応する制

御信号を出力する。T S T B (Task Schedule Table Base) 5 0 4 は、R A M 1 0 2 内に設けられたタスクスケジュールテーブルの先頭アドレスを保持するレジスタである。

【0033】T S T P (Task Schedule Table Pointer) 5 0 5 は、タスクスケジュールテーブルにおいて、タスク切り換え要求が発生した時点で現在実行中のタスクの次に実行すべきタスクを指すポインタを保持するレジスタである。オペランドアドレスバッファ (以下、D O A B と略す) 5 0 6 は、上記退避及び復帰に際してメモリアドレスを保持し、メモリバス中のアドレスバスを介してメモリ1に出力する。

【0034】ストアバッファ (以下、D S T B と略す) 5 0 7 は、上記退避に際して退避すべきデータを保持し、メモリバス4 1 9 中のデータバスを介してメモリ1に出力する。ロードバッファ (以下、D S T B と略す) 5 0 8 は、上記復帰に際して、メモリ1から読み出された復帰すべきデータを、メモリバス4 1 9 中のデータバスを介して入力して保持する。

【0035】加算器5 0 9 は、D O A B 5 0 6 の内容と、定数発生器5 1 0 からの定数加算し、加算結果をD O A B 5 0 6 に出力する。これにより退避時および復帰時に、それぞれの退避領域の個々データ格納場所のアドレスを先頭から順に生成する。図3は、タスクスケジュールテーブル、タスク用退避領域のメモリマップの一例を示す。

【0036】同図において、「T S T B」は、タスクスケジュールテーブルの先頭アドレスを指すポインタであり、T S T B 5 0 4 に保持される。「T S T P」は、タスク切り換え要求が発生した時点で現在実行中のタスクの次に実行すべきタスクを指すポインタあり、T S T P 5 0 5 に保持される。「タスクスケジュールテーブル」は、各タスク用のスタックポインタの退避場所を示すアドレスを、タスクの実行順に保持し、末尾には' F F (16進数) を保持する。同図の例では、4つのタスク1→4が、1→2→3→1→4の順に実行することが指定されている。この例ではタスク1が2回指定されているので、タスク1→4のそれぞれのC P U 占有率は、4 0 %、2 0 %、2 0 %、2 0 %となる。

【0037】「タスク用退避領域」には、「コンテキスト退避領域」と、「スタックポインタ退避領域」とからなる。「コンテキスト退避領域」は、本実施例では各タスク用のスタック領域内に設けられている。つまりスタック領域は、通常のスタックとしての使用に加えてコンテキストの退避領域としても用いられる。同図では、タスク1→4それぞれのコンテキストとして、A 1、A 0、D 1、D 0、P S W、P C (このP C はI A B 4 0 3 またはP F C 4 0 4 の内容) の各レジスタの内容が記憶される。

【0038】「スタックポインタ退避領域」は、各タス

ク毎にスタックポインタの内容を記憶する。同図ではタスク1→4の各スタックポインタの内容が記憶されている。この領域に退避されているスタックポインタの値は、現在実行されていない各タスクについて、コンテキスト退避領域の先頭位置を指す。例えば、退避されたタスク3用スタックポインタの値は、タスク3からタスク1に切り換えられた直後である場合には、タスク3の実行で用いられていたスタック領域の先頭位置を指すが、タスク3のコンテキストの退避が完了した後は、コンテキスト退避領域の先頭位置を指す。さらにその後、タスク3のコンテキストの復帰が完了した後は、タスク3のスタック領域の先頭位置を指す。タスク3の実行中は意味を持たない。

【0039】同図ではタスク3から1への切り換え直後に、タスク3のコンテキストが退避され、次のタスク切り換え要求発生後に実行すべきタスク4のコンテキストが復帰される様子を示している。このとき、コンテキスト退避領域のアドレスはD O A B 5 0 6 により指定され、退避データはD S T B 5 0 7 によりメモリ1に出力され、復帰データはD L D B 5 0 8 によりメモリ1から入力される。

【0040】図4は、転送制御部5 0 1 の機能的な構成を示すブロック図である。転送制御部5 0 1 は、物理的には図2に示したP L A 5 0 2 とマイクロ命令レジスタ5 0 3 から構成されるが、機能的には図4のようによく表すことができる。この転送制御部5 0 1 は、内部バス空き検出回路5 1、メモリバス空き検出回路5 2、レジスタ読み出し制御部5 3、メモリ書き込み制御部5 4、メモリ読み出し制御部5 5、レジスタ書き込み制御部5 6 から構成される。

【0041】内部バス空き検出回路5 1 は、内部バス4 1 8 が空きである (C P U によって使用されていない) サイクルを検出する。より具体的には、内部バス空き検出回路5 1 は、マイクロ命令レジスタ4 1 1 の内部バス制御フィールドがN O P (ノー・オペレーション) を示す場合に空きであると判定するとともに、空きである場合にレジスタ読み出し制御部5 3 及びレジスタ書き込み制御部5 6 にレジスタへのアクセスを許可する。ここで、内部バス制御フィールドとは、マイクロ命令レジスタ4 1 1 から出力される各種制御信号群のうち、内部バスに対するデータ入出力を指示する制御信号からなるフィールドをいう。またN O Pとは、当該フィールド内の全ての制御信号がアクティブでない状態をいう。

【0042】メモリバス空き検出回路5 2 は、メモリバス4 1 9 が空きである (C P U によって使用されていない) サイクルを検出する。より具体的には、メモリバス空き検出回路5 2 は、バスコントローラ4 2 0 のメモリバス制御フィールドがN O P を示す場合に空きであると判定するとともに、空きである場合にメモリ書き込み制御部5 4 及びメモリ読み出し制御部5 5 にメモリへのア

11

クセスを許可する。ここで、メモリバス制御フィールドとは、バスコントローラ420から出力される全ての制御信号フィールドのうち、メモリバスに対するデータ入出力を指示する制御信号からなるフィールドをいう。

【0043】レジスタ読み出し制御部53は、内部バス空き検出回路51によりレジスタへのアクセスが許可されているときに、CPUのタスク実行に使用されていないレジスタセットから個々のレジスタデータを順に読み出してDSTB507に格納する。メモリ書き込み制御部54は、メモリバス空き検出回路52によりメモリ1へのアクセスが許可されているときに、DSTB507に格納されたデータをメモリ1のコンテキスト退避領域に順に書き込む。その際、メモリ書き込み制御部54は、まず、現在実行中のタスクの1つ前のタスクのコンテキスト退避領域の先頭アドレスをDOAB506に設定し、以降、データを書き込む毎にDOAB506を更新する。

【0044】メモリ読み出し制御部55は、コンテキストの退避完了後、メモリバス空き検出回路52によりメモリ1へのアクセスが許可されているときに、メモリ1のコンテキスト退避領域からデータを順に読み出してDLDB508に格納する。その際、メモリ読み出し制御部55は、まず、現在実行中のタスクの次のタスクのコンテキスト退避領域の先頭アドレスをDOAB506に設定し、以降、データを書き込む毎にDOAB506を更新する。

【0045】レジスタ書き込み制御部56は、内部バス空き検出回路51によりレジスタへのアクセスが許可されているときに、DLDB508のデータを、CPUのタスク実行に使用されていないレジスタセットの対応するレジスタを順に書き込む。図5は、CPU4によるタスク切り換え処理、及び、転送部5による詳細なコンテキスト退避・復帰処理を表すフローチャートである。

【0046】まず、タスク切り換え処理について説明する。CPU4は、タスク切り換え要求が発生すると、IAB403またはPFC404の内容をタスク復帰時の戻りPC値として当該タスクのスタック領域に退避するとともに、PSWの内容も退避し（ステップ501）、タスク実行で使用していたレジスタセットを他方のレジスタセットに切り換える、転送制御部501を起動する（ステップ502）。このとき、新たなレジスタセットには次に実行すべきタスクのコンテキストが復帰されているので、CPU4は、次タスクのスタック領域からPSWと戻りPC値とを復帰させることにより次タスクの実行を開始する（ステップ503）。

【0047】次に、転送部5によるコンテキスト退避・復帰処理を説明する。なお、フローチャート中“1”を記した各ステップは、内部バス空き検出回路51により内部バスが空いていると検出されたサイクルにおいて実行されることを示し、“M”を記した各ステップは、メモ

12

リバス空き検出回路52によりメモリバスが空いていると検出されたサイクルにおいて実行されることを示す。

【0048】転送部5の起動後、レジスタ読み出し制御部53は、ステップ502の切り換えによりCPUに使用されなくなったレジスタセットのSPの内容を読み出し、DOAB506に格納する（ステップ504）。このときDOAB506の内容は、スタック領域内に設けられたコンテキスト退避領域の先頭アドレスを表す。レジスタ読み出し制御部53およびメモリ書き込み制御部54は、レジスタセットの各レジスタの内容をコンテキスト退避領域に退避する（ステップ505～508）。さらに、メモリ読み出し制御部55及びメモリ書き込み制御部54は、この時点のDOAB506の内容をスタックポイント退避領域に退避する（ステップ509～511）。これにより退避後のコンテキスト退避領域の先頭アドレスがスタックポイント退避領域に退避される。これでコンテキストの退避が完了する。

【0049】次に、メモリ読み出し制御部55は、TSTP505の内容を現在実行中の次に実行すべきタスクを指すように更新し（ステップ512）、次タスクのコンテキスト退避領域の先頭アドレスをスタックポイント退避領域から読み出し、DOAB506に格納する（ステップ513）。このときDOAB506の内容がx'FF（16進数）であれば、テーブルの末尾を表すのでTSTP505の内容をTSTB504に保持された先頭アドレスに更新してから再度ステップ513を行う（ステップ514、515）。

【0050】さらに、メモリ読み出し制御部55及びレジスタ書き込み制御部56は、次タスクのコンテキスト退避領域の各データを、レジスタセットに復帰する（ステップ516～519）。さらに、レジスタ書き込み制御部56は、この時点のDOAB506の内容をレジスタセットのSPに書き込む（ステップ520）。これにより次タスクのスタック領域の先頭アドレスがSPに設定される。これで次タスクのコンテキストの復帰が完了する。

【0051】以上のように構成された本発明の実施例におけるマイクロコンピュータについて、タスク切り換え動作を説明する。今、タスクスケジュールテーブル2は、図3に示したように4つのタスク1～4が、1→2→3→4の順に実行することが指定されていて、タスク3が実行中であるものとする。この状態でタイマー割込みが、タスク切り換えを要求するイベントとして発生した場合、CPU4は、割込み処理においてタスク3の戻りPC及びPSWの退避、レジスタセットの切り換え、タスク1のPSW及び戻りPCの復帰（図5のステップ501～503）をすることにより、タスク3からタスク1に切り換える。

【0052】転送部5は、レジスタセットの切り換えと同時に起動され、CPU4のタスク1実行中に内部バス

50

418、メモリバス419の空きサイクルを利用して、図3および図5に示したように、タスク3のコンテキストの退避処理と、タスク4のコンテキスト復帰処理とを行なう。以上説明してきたように本発明のマイクロコンピュータによれば、CPU4は、戻りPC及びPSWの退避・復帰と、レジスタセットの切り換えを行なうだけでタスクを切り換えることができるので、タスク切り換えによるCPUのオーバーヘッドが極めて少なく、高速なタスク切り換えを実現する。しかも、タスク切り換えに必要なハードウェア構成は、レジスタセットを1つ分と転送部を追加するだけなので、より小さいハードウェア規模で実現できるという利点がある。この点、特に組み込み用途の1チップマイクロコンピュータに適している。

【0053】また転送部5は、CPU4のタスク実行中に内部バス及びメモリバスの空きサイクルを利用してコンテキストの退避・復帰を行なうので、CPUのタスク実行を全く妨げないので、CPUの処理能力を最大限に活用することができる。さらに、2つのレジスタセットを交互にタスクに割当るので、CPUが実行するタスク数に制限がない。このタスク数は、コンテキスト退避領域を設けることが前提なので、メモリ容量の許す限りタスク数を増やすことができる。

【0054】また、タスクスケジュールテーブル2内に同一タスクを複数記述するだけで、タスクのCPU占有時間の制御を容易にすることができます。例えば、図3の例ではタスク1は、テーブル内の2箇所に記述されるだけで、他のタスク2、3、4の2倍の占有時間(40%)を確保している。なお、上記実施例では、タスク切り換えを要求するイベントを、周期200mSのタイマー割込とした。イベント周期は、転送部5がCPU4のタスク実行の背後でバスの空きサイクルを利用してコンテキストの転送(退避・復帰)を行なっているので、タスク実行においてバスの空きサイクル数が十分生じるだけの時間を確保する必要がある。本実施例の構成ではイベント周期として数百μS～数mS以上が望ましい。

【0055】また、上記実施例では、イベントをタイマー割込としたが、これ以外にもタスクスイッチ命令、特定ポート(シリアルポートなど)の割り込み等であってもよい。上記実施例では、図3に示したようにタスク退避領域をSP退避領域とコンテキスト退避領域とに分けて構成したが、1つの領域にまとめてよい。また、タスクスケジュールテーブル2にはタスク毎にSP退避領域のアドレスを格納するよう構成したが、タスク番号を格納する構成としてもよい。この場合、タスク番号に対応するSP退避領域のアドレスを求めるためのテーブルを設ければよい。

【0056】図8は、本発明の他の実施例におけるマイクロコンピュータの詳細な構成を占めずブロック図である。同図の構成は、図2に対してクロック供給部421の代わりにクロック供給部422が設けられている点

と、停止回路511が追加されている点のみが異なる。これらの点以外は先の実施例と同じなので異なる点のみを説明する。

【0057】クロック供給部422は、CPU4および転送部5に動作クロックを供給する点は同じであるが、CPU4に対しては停止部511の指示に従って動作クロックの供給を停止する点が異なる。停止部511は、転送部5の転送動作中に、タスク切り換えを要求するイベントが発生した場合に、転送動作が終了するまでの間クロック供給部422によるCPU4へのクロック供給を停止する。具体的には、転送部5が転送動作中であることを示す転送中フラグを内部に有し、退避開始時点でセットし、復帰終了時点でリセットする。また、停止部511は、前記イベントを表すタイマー割り込み信号がアクティブで、かつ、フラグがセットされていれば、クロック供給部422がCPU4への動作クロック供給を停止するよう指示する。その後フラグがリセットされた時点で、前記停止指示を解除する。

【0058】以上のように構成された本実施例のマイクロコンピュータの動作を以下説明する。転送部5により退避・復帰動作は先の実施例と全く同じである。この退避・復帰動作中に、タスク切り換えを要求するタイマー割込みが発生したものとする。この場合、停止部511の指示によってクロック供給部422がCPU4への動作クロックの供給を停止する。これによりCPU4は、その動作を停止する。転送部5への動作クロックは供給されているので、復帰動作の完了後には、前記フラグがリセットされる。これにより停止部511は、前記停止指示を解除するので、CPU4によるタスク切り換えおよびタスク実行の動作が再開される。

【0059】以上のように本実施例では、転送部5による転送動作が終了しないうちに、タスク切り換えを要求するイベントが発生した場合には、CPU4の動作を止めて転送を完了させてからCPU4の動作を再開させる。これにより、転送部5の転送時間がイベント発生間隔よりも短い場合に、適切にタスク切り換えを実行することができる。例えば、イベントが周期的な割込みではなく不定期に発生する割込みである場合や、内部バス418、メモリバス419の空きサイクルがほとんど発生しないようなタスク(メモリデータの転送が大部分であるようなタスク)が存在する場合に、動作が停止する分だけCPU4の処理能力が低くなるが、適切にタスク切り換えを実行することができる。

【0060】
【発明の効果】以上説明したように、請求項1の発明によれば、レジスタセットを1つ分と、転送手段をハードウェアとして追加するだけで、高速なタスク切り換えを実現することができるという効果がある。特に組み込み用途のマイクロコンピュータにおいては、ハードウェア

規模の制約があっても、タスクの高速切り換えを実現で

きるので、組み込み先の装置におけるリアルタイム制御を可能にする。

【0061】請求項2の発明によれば、請求項1の効果に加えて、タスク実行を全く妨げることなく、退避と復帰を実現できるという効果がある。請求項3の発明によれば、レジスタセットを1つと、転送手段をハードウェアとして追加するだけで、高速なタスク切り換えを実現することができるという効果がある。特に組み込み用途のマイクロコンピュータにおいては、ハードウェア規模の制約があっても、タスクの高速切り換えを実現できるので、組み込み先の装置におけるリアルタイム制御を可能にする。

【0062】請求項4の発明によれば、請求項3の効果に加えて、タスクの実行順序として1つのタスクを複数指定するだけで、中央演算処理手段におけるタスクの占有率を簡単に制御できるという効果がある。請求項5の発明によれば、請求項3又は4の効果に加えて、タスク実行を全く妨げることなく、退避と復帰を実現できるという効果がある。

【0063】請求項6の発明によれば、請求項3、4又は5の効果に加えて、退避・復帰に要するよりも短い周期でタスク切り換えの要求が発生しても適切にタスクを切り換えることができるので、タスク切り換え周期を幅広く設定することができる。

【図面の簡単な説明】

【図1】本発明の実施例におけるマイクロコンピュータの主要な概略構成を示すブロック図である。

【図2】同実施例におけるマイクロコンピュータのより詳細な構成を示すブロック図である。

【図3】同実施例におけるタスクスケジュールテーブル、タスク用退避領域のメモリマップの一例を示す。

【図4】同実施例における転送制御部501の機能的な構成を示すブロック図である。

【図5】同実施例におけるCPU4によるタスク切り換*

* え処理、及び、転送部5による詳細なコンテキスト退避・復帰処理を表すフローチャートである。

【図6】第1の従来技術におけるマルチタスク機能を有するコンピュータの説明図である。

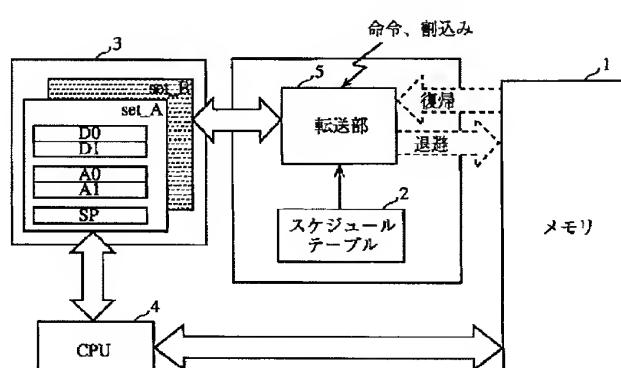
【図7】第2の従来技術におけるマルチタスク機能を有するコンピュータの説明図である。

【図8】本発明の他の実施例におけるマイクロコンピュータの詳細は構成を示すブロック図である。

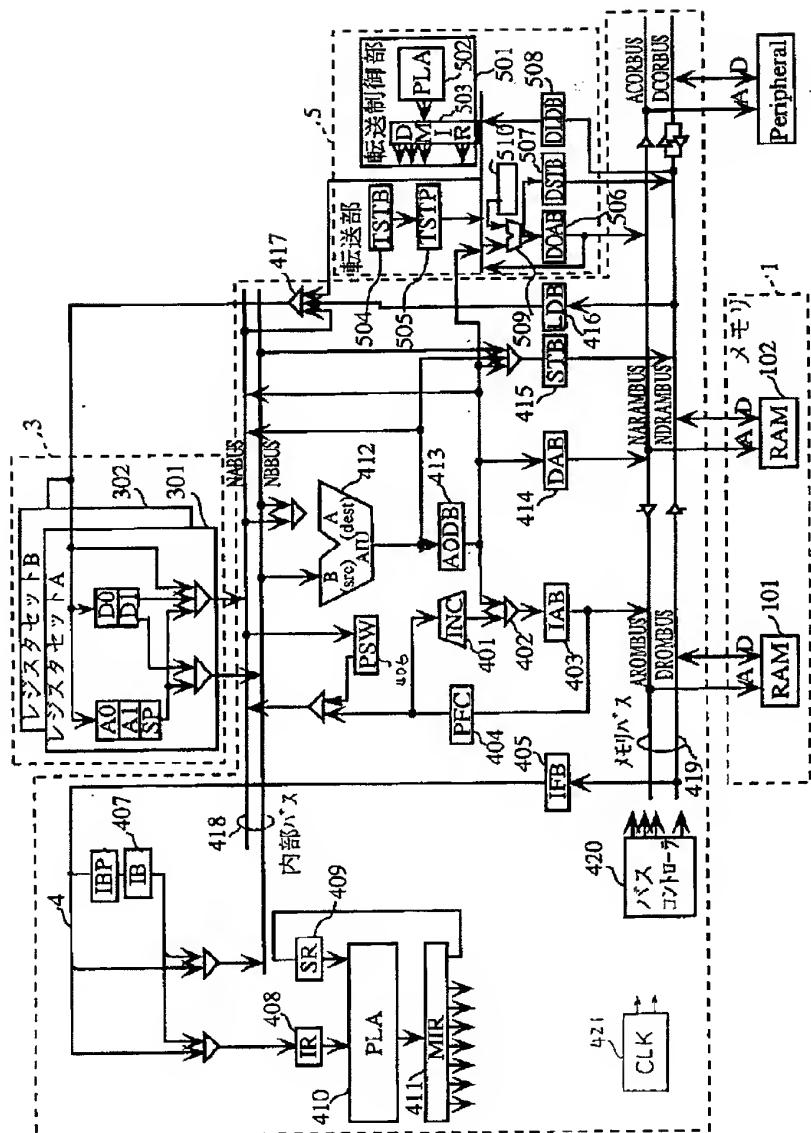
【符号の説明】

10	1	メモリ
	2	タスクスケジュールテーブル
	3	レジスタファイル
	4	CPU
	5	転送部
20	5 1	内部バス空き検出回路
	5 2	メモリバス空き検出回路
	5 3	レジスタ読み出し制御部
	5 4	メモリ書き込み制御部
	5 5	メモリ読み出し制御部
	5 6	レジスタ書き込み制御部
	4 2 2	クロック供給部
	5 0 1	転送制御部
	5 0 2	PLA
	5 0 3	マイクロ命令レジスタ
	5 0 4	TSTB
	5 0 4	ステップ
	5 0 5	TSTP
	5 0 6	DOAB
	5 0 7	DSTB
30	5 0 8	DLDB
	5 0 9	加算器
	5 1 0	定数発生器
	5 1 1	停止部

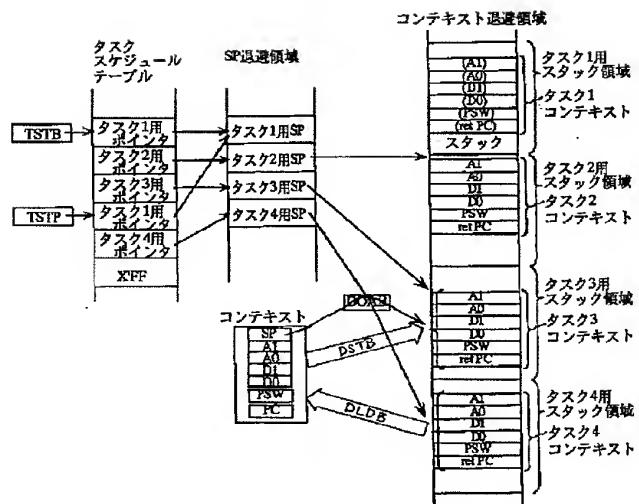
【図1】



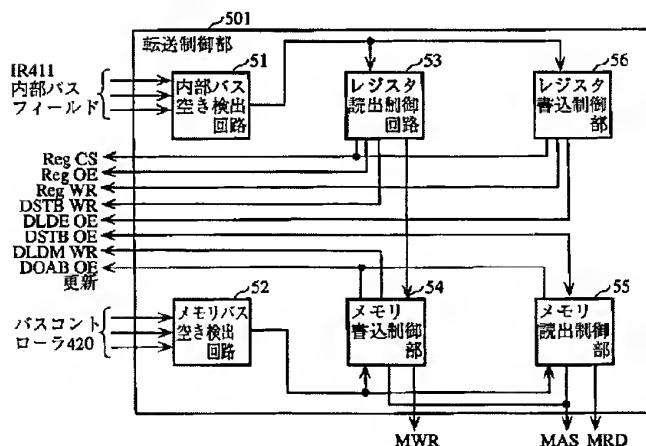
[図2]



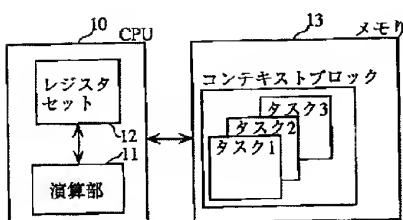
【図3】



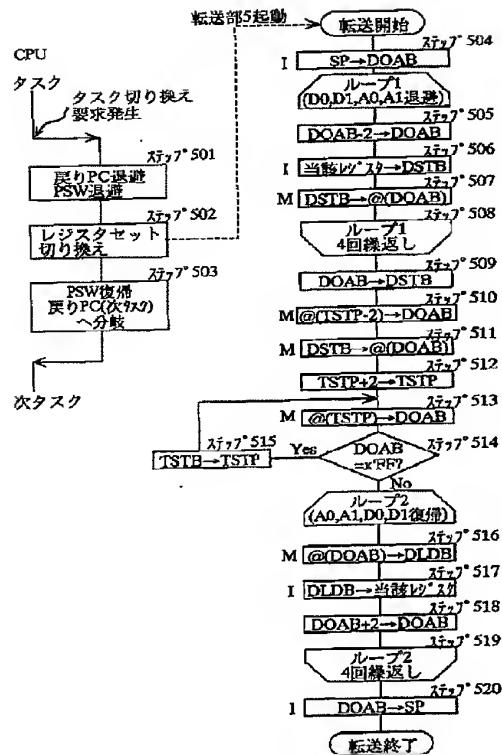
【図4】



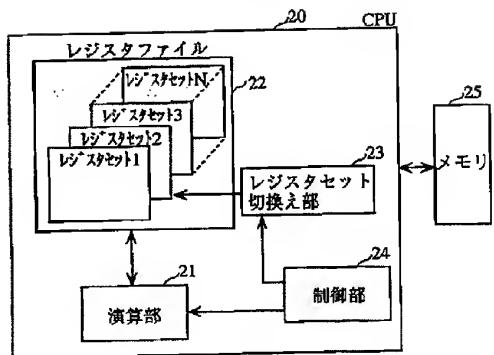
【図6】



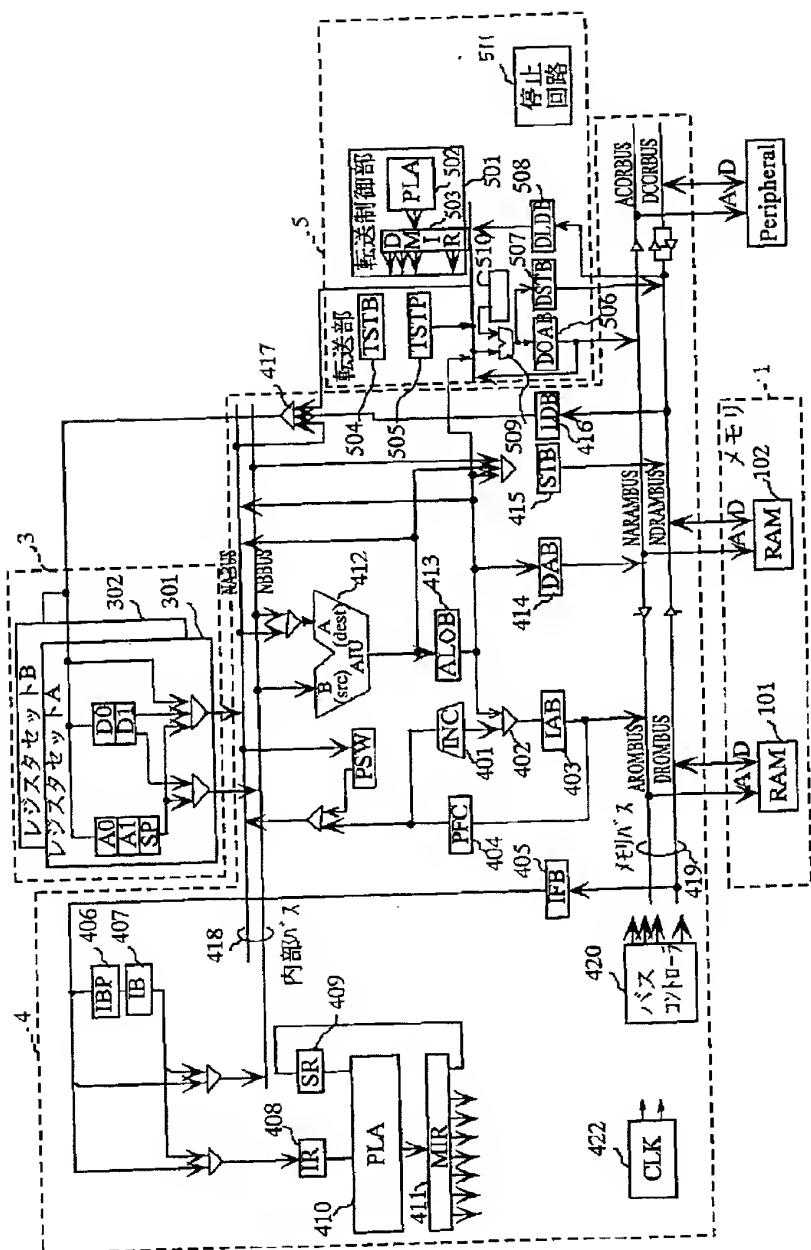
【図5】



【図7】



[図8]



フロントページの続き

(72)発明者 松本 政彦
大阪府門真市大字門真1006番地 松下電器
産業株式会社内